REMARKS

In the Office Action, claims 1-31 are pending in the case. The Examiner rejected all the claims under 35 U.S.C. §102(e). By this paper, Applicant has responded to each of these rejections, made minor corrections to the specification, and amended the independent claims.

THE ABSTRACT

Applicant has amended the abstract to provide a more succinct abstract of one rather than two paragraphs.

THE SPECIFICATION:

Applicant has amended the specification to correct a spelling error.

THE CLAIMS:

Amendments

Applicant has amended claim 1 to include a merge end point utility which identifies a merge end point to distinguish between spill records and detail records in a log. Also, claim 1 has been amended to clarify that a change accumulation manager reads a change accumulation data set to derive detail records.

Claims 11 and 21 have been amended to include the step of determining a merge end point and reading a log to derive updates subsequent to a merge end point and applying the updates to the restored database data set. Applicant asserts that such a step more accurately reflects the steps of the invention.

Claims 11 and 21 have also been amended to clarify that the step of "reading a change accumulation data set" is performed "to derive detail records associated with the database data set." Support for these amendments is found in the specification at page 9, lines 8- 9, lines 21-22 and page 10, lines 1-2; page 22 lines 3-9; page 23 line 17 through page 24 line 10.

35 U.S.C. 102(e)

The Examiner rejected claims 1-31 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,397,308 to Yuval Ofek (hereinafter "Ofek"). Applicant has amended independent claims 1, 11, and 21 and appropriate dependent claims to further clarify the invention.

Specifically, claim 1, as amended, includes "a merge end point utility configured to determine the merge end point reflective of a separation of detail and spill records in a change accumulation data set." See *amended claim 1.*

As described in the specification, conventional restoration of a database using change accumulation data sets (CADS) involves a two stage process. In a first stage, CADS which are incomplete are finished by reading in all applicable CADS and logs. The incomplete CADS are then converted into complete CADS by identifying the spill records which belong together. Combining the spill records forms a completed transaction. The complete CADS are then used in the second stage with a backup copy of the database and log records to restore the database. The process is lengthy, particularly where multiple CADS, and logs are involved, because the CADS and backup copy must be read sequentially. See *specification* page 7, line 9 - page 8, line 16.

However, in the claimed invention the CADS are read in parallel with the reading and restoring of the backup copy and restoration from the logs. This parallel operation is made possible because a merge end point is identified for the CADS and the logs. See *specification* page 20, lines 13-21; page 22, lines 3-5. A merge end point separates the updates in the CADS and logs into detail records and spill records. Because the merge end point is known, all the detail records up to the merge end point may simply be applied to the backup database copy being restored when needed. Next, depending on the embodiment, the spill records may either be applied to the database or ignored. See *specification* page 24, lines 5-10.

Identification of a merge end point allows for the CADS and logs to be processed in parallel and on a single pass. In addition, detail records are applied in parallel. This means there is no separate stage for processing one or more CADS to form complete CADS. This results in a time savings for the database recovery process.

In addition, because CADS are created periodically from logs, only a subset of the existing logs need to be read and analyzed to do a database restoration. This too can greatly reduce the time needed to perform a database recovery.

Claims 11 and 21 include the step of "reading a log to derive updates subsequent to a merge end point and applying the updates to the restored database data set." See *amended claims 11 and 21*. The step of "reading a change accumulation data set..." has been amended to clarify that the reading of the change accumulation data set is to derive detail records. This step is possible because a merge end point is identified and provides similar benefits to those described above.

The step of "reading a log to derive updates..." further clarifies the inventive method. The log is read to identify any updates which occurred after the latest CADS was created but before the time of data base failure. By reading a log, the recovery process can restore the data within the database up to substantially the point of failure.

In order for an invention to be anticipated under 35 U.S.C. § 102, "[E]very element of the claimed invention must be identically shown in a single reference." *In re Bond*, 910 F.2d 831, 15 USPQ 2d 1566 (Fed. Cir. 1990). With regard to method claims, anticipation requires identity of the claimed process and a process of the prior art; the claimed process, including each step thereof, must have been described or embodied, either expressly or inherently, in a single reference. *Scripps Clinic & Research Found. v. Genetech Inc.*, 927 F.2d 1565, 1576, 18 USPQ2d 1001, 1010 (Fed. Cir. 1991); *see Standard Havens Prods., Inc. v. Gencor Indus., Inc.*, 953 F.2d 1360, 1369, 21 USPQ2d 1321, 1328 (Fed. Cir. 1991). Applicant respectfully asserts that Ofek fails to identically show each element and fails to show each step set out in the amended independent claims 1, 11, and 21.

Merge End Point Utility

The Examiner suggests that the claim limitations added to claims 1, 11, and 21, specifically a "merge end point utility" and the step of "reading a log to derive updates subsequent to a merge end point..." are identically shown in Ofek. The Applicant respectfully disagrees. The Examiner points to a process, in Ofek, for backing up data from a primary physical storage space used by a database. The database is placed in "on-line backup mode." While the primary physical storage space is being backed up the updates are stored in a "re-do" log. See *Ofek* column 27 – column 28, line 68.

A "re-do" log is fundamentally different from a conventional log. The re-do log is specifically for use when the primary storage space in unavailable. A conventional database log is a component for use in providing database recovery. The re-do log is a log which may be used to allow a database to remain online while a backup of the primary physical storage space is performed. Therefore, re-do logs are configured to be temporary. Conversely, the database recovery logs are more permanent. Re-do logs quickly fill up and exceed the space allocated to the them. Conventional database recovery logs are configured to automatically allocate new space or switch to create a new log when an existing log fills up.

Furthermore, re-do logs must be read and applied subsequent to the backup operation described in Ofek. This process may be done while the database is online as described in Ofek. However, application of the updates in a conventional database recovery log need only occur if there is a database failure. In fact, because logs may be merged into CADS, some logs may not ever need to be read to properly recover a database. Instead, the CADS may be used.

Therefore, because a backup process for primary storage space which allows a database to remain online is fundamentally different from the safe guard logs which are being used continually by a database system, Applicant submits that the claim language of a "merge end point utility" and the step of "reading a log to derive updates subsequent to a merge end point..." are not identically shown in Ofek. Consequently, amended claims 1, 11, and 21 and their corresponding dependent claims are in condition for immediate allowance.

Change Accumulation Manager

Claim 1 as amended recites among other elements:

"a change accumulation manager configured to read the change accumulation data set to derive detail records in parallel with the read and restore of the backup copy."

Applicant respectfully asserts that change accumulation manager as described in the claimed invention is not found in Ofek.

The Examiner cites to a differential backup process in Ofek. The differential backup process of Ofek tracks changes in a logical construct. This process works with segments which "may be defined by how the data is physically stored." See *Ofek* Column 29, lines 27-28. These segments are blocks of data which together form the logical construct. Ofek then describes a process by which the data blocks are stored on a storage medium such as secondary storage. Applicant asserts that Ofek fails to show a change accumulation manager which derives records in parallel with reading and restoring a backup copy. In addition, the segments referred to in Ofek are not detail records as recited in the claims.

Ofek does not show derivation of detail records from CADS. Detail records are written in CADS and comprise "[U]pdates 402 which are confirmed as being committed." See *specification* page 22, lines 10-11. Those of skill in the art of database design and recovery processes recognize that these detail records are generally different from actual records which may be stored in a database. Rather than a direct copy of a record in the database, a detail record within CADS represents a database operation. Ofek describes how a copy of the actual data comprising a conventional database record may be backed up. However, Ofek fails to discuss recovery and restoration of a database through use of detail records derived from CADS.

Generally, the detail records record what changes to the database are taking place with each update within a transaction. A transaction is generally defined as one or more updates which access or change the contents of a database. The detail records record an update associated with a transaction, not simply the data of a database record. Generally, the detail record includes an indicator of whether a database update is to be removed or applied to a database. If the update is to be removed, the original value, which is stored in the detail record is applied to the database. If the update is to be applied, the new value is applied to the database.

Detail records within the CADS taken together represent one or more completed transactions. A transaction is one or more updates which where committed to the database, actually written in physical storage.

For example, if the transaction is a database operation for changing a bank account balance (*i.e.* making a withdrawal) many update commands may be issued. First, the user may be authenticated. Next, the current balance may be checked to ensure available funds. And finally, the account may be debited the proper amount. One or more detail records for each of the update commands is generally recorded in a database log. Periodically, the database logs in a shared database environment are compiled into one or more incomplete CADS. See *specification*, page 6, lines 6-14. Each detail record may indicate whether an update is to be applied or removed, the original account value and the new account value. Each detail record may also include an identifier such as an offset into the database file. This identifier uniquely identifies the detail record and indicates where in the database file the change is to be made.

The change accumulation manager derives the detail records from the CADS. Generally, derivation of the detail records involves reading each detail record and identifying whether to apply or back out the change represented by the detail record. In certain implementations, the detail record contains the data values which are to be removed and replaced with an original value. Thus, the change is backed out. Alternatively, all detail records may be simply applied to the database. Backing out updates may require that twice as many detail records be stored in a CADS. The first set may change the database and the second database may be an inverse of the first set which consequently backs out the change. Which implementation is used is not critical to the invention. The detail records themselves must be analyzed. This is not simply a process of copying data bit for bit as disclosed in Ofek.

Furthermore, the change accumulation manager responds to queries from the backup copy restore utility. The queries are for detail records which have been read by the change accumulation manager. The backup copy restore utility sends the queries when database restoration process comes to a position in the file where any existing detail records may be applicable. See *specification*, page 25, lines 1-13. Applicant finds no references to a change accumulation manager which reads a CADS and responds to queries from a backup copy restore

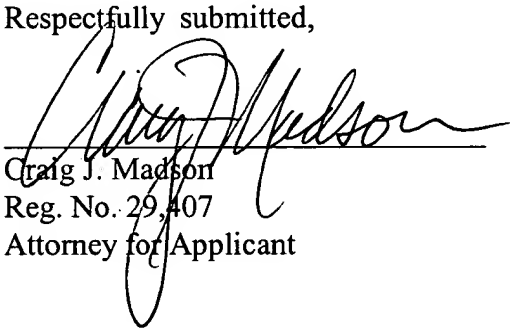
utility. Therefore, Applicant asserts that Ofek does not identically show every element, specifically the change accumulation manager, of the claimed invention.

Applicant asserts that claims 2-3, and 5-10 are allowable as being dependent on an allowable base claim for the reasons stated above. In addition, independent claims 11 and 21 are directed to substantially the same subject matter as claim 1. Therefore, for similar reasons, Applicant asserts that claims 12-13, 15-20, 22-24, and 26-31 are allowable as being dependent on an allowable base claims.

In view of the foregoing, Applicant submits that the application is in condition for immediate allowance. In the event any questions remain, the Examiner is respectfully requested to initiate a telephone conference with the undersigned.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned **"Version With Markings To Show Changes Made."**

Respectfully submitted,



Craig J. Madson
Reg. No. 29,407
Attorney for Applicant

Date: September 18, 2002

MADSON & METCALF
Gateway Tower West
15 West South Temple, Suite 900
Salt Lake City, Utah 84101
Telephone: 801/537-1700

S:\ALLCLIENTS\3000 IBM\3000.2.12\OA1\3000.2.12OAR.doc

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE ABSTRACT

The abstract has been amended as follows:

~~The present invention is a~~A recovery utility apparatus for expediting recovery time during failure of one or more database data sets is provided. A backup copy restore utility is configured to reads and restore a backup copy of the database data set. A change accumulation manager is configured to read detail records in an incomplete change accumulation data set. A log manager is configured to read a log to derive updates subsequent to a merge end point; wherein the updates are reflective of spill records. An image copy restore utility is configured to apply the detail records to the backup copy to thereby create a restored database data set. A database update manager is configured to apply the updates to the restored database data set.~~one or more backup copies of the database data sets in parallel. Simultaneously, a change accumulation manager reads one or more CADSs in parallel. Each CADSs associated with one or more database data sets requiring recovery is only read once into memory. In this manner, parallel execution of the read process reduces recovery time. To further expedite recovery, as the backup copy is written to the restored database, records from the CADS are merged with the restored database as they are needed and as they become available.~~

~~The change accumulation manager reads only the detail records which have been committed and ignores the spill records to eliminate the need for completing each incomplete CADS for recovery. A log manager reads one or more logs to derive the updates in the spill records which are subsequent to a merge end point. Reading the logs confirms which updates in the spill records have been committed and may be merged with the restored database. The logs~~

~~are read in parallel to reduce read time and are merged with the restored database before the read process is complete.~~

IN THE SPECIFICATION:

On page 5, please replace the paragraph beginning on line 19 and continuing onto page 6 with the following paragraph:

Each CADS comprises a detail record which is a record of committed updates from one or more logs. Each detail record is a ~~serious~~ series of contiguous bytes which can be overlaid into the backup copy of one database physical record. Applying all of the detail records in the CADS is equivalent to rerunning all of the transactions against the data base which were entered since a backup copy was made up to a "merge-end point." The merge-end point is a point in time wherein updates may no longer be merged with the new database because all change records are not available for these updates. Thus, there is no guarantee as to whether these updates have been committed. Updates which cannot be merged with the new database are written to records which are termed "spill records."

On page 5, please replace the paragraph beginning on line 19 and continuing onto page 6 with the following paragraph:

In step 512, the image copy and restore utility 316 queries the CADS manager 304 as to whether a specific detail record 404 required for the restored database 318 has been read yet. As detail records 406 are read by the CADS manager 304 into memory 18 the records ~~404~~ 406 are sent to the image copy and restore utility 316 as requested. If there is a delay in the request for the detail records 406 some or all of the detail records 406 may be stored on the virtual memory 25 for longer term storage.

IN THE CLAIMS:

Please cancel claims 4, 14, and 25.

1. (Amended) An apparatus for recovering a failed database data set, the apparatus comprising:

a memory device storing executable modules, the modules comprising:

a recovery utility having,

a backup copy restore utility configured to read and restore a backup copy of the database data set;~~,~~

a merge end point utility configured to determine the merge end point reflective of a separation of detail and spill records in a log;

a change accumulation manager configured to read ~~a~~ the change accumulation data set to derive detail records in parallel with the read and restore of the backup copy ~~to derive detail records,~~; and

an image copy restore utility configured to apply the detail records to the backup copy during the read and restore of the backup copy to thereby create a restored database data set.

11. (Amended) A method for recovering a failed database data set, the method comprising:

reading and restoring a backup copy of the database data set;

reading a change accumulation data set to derive detail records associated with the database data set in parallel with the reading and restoring of the backup copy ~~to derive detail records associated with the database data set~~; ~~and~~

applying the detail records to the backup copy during the reading and restoring of the backup copy to thereby create a restored database data set; and ~~.~~

determining a merge end point within a log and reading the log to derive updates subsequent to the merge end point and applying the updates to the restored database data set.

15. (Amended) The method of claim ~~14~~ 11 wherein reading the log and applying the updates are executed after restoring the backup copy.

21. (Amended) A computer readable medium having stored thereon computer executable instructions for performing a method for recovering a failed database data, the method comprising:

reading and restoring a backup copy of the database data set;

reading a change accumulation data set to derive detail records associated with the database data set in parallel with the reading and restoring of the backup copy ~~to derive detail records associated with the database data set~~; ~~and~~

applying the detail records to the backup copy simultaneously during the reading and restoring of the backup copy to thereby create a restored database data set; and ~~.~~

determining a merge end point within a log and reading the log to derive updates

subsequent to the merge end point and applying the updates to the restored database data set.

26. (Amended) The computer readable medium of claim ~~25~~ 21 wherein reading the log and applying the updates are executed after restoring the backup copy.